REMARKS

In the Official Action mailed on **10 May 2005**, the Examiner reviewed claims 1-27. Claims 1-27 were rejected under 35 U.S.C. §103(a) as being anticipated by Litzkow et al (*Checkpoint and Migration of UNIX Processes in the Condor Distributed Processing System*, hereinafter "Litzkow"), in view of Croix (USPub 2002/0100034, hereinafter "Croix").

Rejections under 35 U.S.C. §102(b)

Claims 1-27 were rejected as being anticipated by Litzkow in view of Croix. Applicant respectfully points out that the combined system of Litzkow and Croix teaches **re-linking submitted programs** to include a checkpointing library and then calling the routines in the checkpointing library by using system calls (see Litzkow, page 1, section 1 and page 5-6, section 3.4.1). Note that Litzkow teaches that a **major limitation** is that the checkpointing code must be linked in with the user's code (see Litzkow, section 4.2, last paragraph).

In contrast, the present invention dynamically links an interceptor library into the application **at application startup time** (see page 6, line 25 to page 7, line 6 of the instant application). This is beneficial because the programmer does not need to perform a linking operation. This enables an interceptor library to be attached to a program at program startup time by simply setting an environment variable. Hence, any application, even one that is already compiled and linked, can be checkpointed using the present invention. this is an important advantage, because a person who wants to checkpoint an application may not have access to the source code or software modules required to link the application. Hence, this invention makes it possible to checkpoint accounting software, even though the programmer does not have access to source code or link modules for the accounting software. In contrast, Litzkow requires the programmer to have access to source code or linkable modules for the application.

There is nothing within Litzkow, either explicit or implicit, which suggests linking the interceptor library at application startup time by setting an environment variable and then calling the system routines from the interceptor library by using pointers gathered by and saved in the interceptor library.

Accordingly, Applicant has amended independent claims 1, 10, and 19 to clarify that the present invention dynamically links an interceptor library into the application at application startup time. These amendments find support on page 6, line 25 to page 7, line 6 of the instant application.

Hence, Applicant respectfully submits that independent claims 1, 10, and 19 as presently amended are in condition for allowance. Applicant also submits that claims 2-9, which depend upon claim 1, claims 11-18, which depend upon claim 10, and claims 20-27, which depend upon claim 19, are for the same reasons in condition for allowance and for reasons of the unique combinations recited in such claims.

CONCLUSION

It is submitted that the present application is presently in form for allowance. Such action is respectfully requested.

Respectfully submitted,

By 

Edward J. Grundler
Registration No. 47,615

Date: 16 June 2005

Edward J. Grundler
PARK, VAUGHAN & FLEMING LLP
2820 Fifth Street
Davis, CA 95616-7759
Tel: (530) 759-1663
FAX: (530) 759-1665